第3章 MATLAB的数值计算

本章主要介绍多项式的运算和方程式的求解。

3.1 多项式

多项式在代数中占有重要的地位，广泛用于数据插值、数据拟合和信号与系统等应用领域。MATLAB 提供了各种多项式的运算方法，使用起来非常简单方便。

3.1.1 多项式的运算

多项式通式可以表示为

$$f(x)=a_n x^n+a_{n-1}x^{n-1}+\cdots+a_1 x^1+a_0$$

多项式之间可以进行四则运算，其结果仍为多项式。在 MATLAB 中，用多项式系数向量进行四则运算，得到的结果仍为多项式系数向量。

1. 多项式的加减运算

MATLAB 没有提供多项式加减运算的函数。事实上，多项式的加减运算是合并同类型的运算，可以用多项式系数向量的加减运算实现。如果多项式阶次不同，则把低次多项式系数不足的高次项用 0 补足，使多项式系数矩阵具有相同维度，以便进行加减运算。

poly2sym 函数可以把输入的系数转换成符号多项式，而 sym2poly 函数为 poly2sym 函数的逆函数，它可以将多项式转换成系数。

语法格式：

```
syms  x;              %定义 x 为符号变量
p=[an  …a1  a0]       %输入多项式系数
fx= poly2sym(p,x)     %由系数转换为多项式
p= sym2poly(fx)       %由多项式转换为系数
```

2. 多项式的乘法运算

在 MATLAB 中，两个多项式的乘积运算可以用函数 conv 实现，其调用格式为

```
p = conv(p1,p2)
```

其中，p1 和 p2 是两个多项式的系数向量；p 是两个多项式乘积的系数向量。

3. 多项式的除法运算

在 MATLAB 中，两个多项式的除法运算可以用函数 deconv 实现，其调用格式为

```
[q,r] = deconv(p1,p2)
```

其中，q 为多项式 p1 除以 p2 的商式；r 为多项式 p1 除以 p2 的余式。q 和 r 都是多项式系数向量。

deconv 函数是 conv 函数的逆函数，即满足 p1＝conv(p2,q)＋r。

【例 3-1】 已知两个多项式：$f(x)=x^4+2x^3+2x^2+x+6$，$g(x)=2x^3-2x^2+1$。

(1) 求两个多项式相加和相减的结果。

(2) 求两个多项式相乘和相除的结果。

程序代码：

```
p1 = [1 2 2 1 6];
p2 = [0 2 2 0 1];
p3 = [2 2 0 1];
p = p1 + p2
poly2sym(p) % 多项式系数表示成符号表达式
p = p1 - p2
poly2sym(p)
p = conv(p1,p2)
poly2sym(p)
[q,r] = deconv(p1,p3)
p4 = conv(q,p3) + r % 验证 deconv 是 conv 的逆函数
```

运行结果：

```
p =
     1     4     4     1     7
ans =
x^4 + 4*x^3 + 4*x^2 + x + 7
p =
     1     0     0     1     5
ans =
x^4 + x + 5
p =
     0     2     6     8     7    16    14     1     6
ans =
2*x^7 + 6*x^6 + 8*x^5 + 7*x^4 + 16*x^3 + 14*x^2 + x + 6
q =
    0.5000    0.5000
```

```
r =
         0         0    1.0000    0.5000    5.5000
p4 =
     1     2     2     1     6
```

下面进行符号多项式和多项式系数之间的相互转化。

程序代码：

```
syms x;
p = 2 * x^7 + 6 * x^6 + 8 * x^5 + 7 * x^4 + 16 * x^3 + 14 * x^2 + x + 6;
p1 = sym2poly(p)
p2 = poly2sym(p1)
```

运行结果：

```
p1 =

     2     6     8     7    16    14     1     6
p2 =

2 * x^7 + 6 * x^6 + 8 * x^5 + 7 * x^4 + 16 * x^3 + 14 * x^2 + x + 6
```

3.1.2 多项式的值和根

1. 多项式的值

在 MATLAB 中，求多项式的值可以用 polyval 函数和 polyvalm 函数。它们的输入参数都是多项式系数和自变量，区别是前者是代数多项式求值，后者是矩阵多项式求值。

1）代数多项式求值

polyval 函数可以求代数多项式的值，其调用格式为

```
y = polyval(p,x)
```

其中，p 为多项式的系数；x 为自变量。若 x 为一个数值，则求多项式在该点的值；若 x 为向量或矩阵，则对向量或矩阵的每个元素求多项式的值。

【例 3-2】 已知多项式 $f(x)=x^3+3x^2+2x+7$，分别求 $x=2$ 和 $\boldsymbol{x}=[0,2,4,6,8,10]$ 时的多项式的值。

程序代码：

```
x1 = 2;
x = [0:2:10];
p = [1 3 2 7];
y1 = polyval(p,x1)
y = polyval(p,x)
```

运行结果：

```
y1 =
    31
y =
           7          31         127         343         727        1327
```

2）矩阵多项式求值

polyvalm 函数以矩阵为自变量求多项式的值，其调用格式为

```
Y = polyvalm(p,X)
```

其中，p 为多项式系数；X 为自变量，要求为方阵。在 MATLAB 中，用 polyvalm 函数和 polyval 函数求多项式的值是不一样的，因为运算规则不一样。例如，假设 A 为方阵，p 为多项式 x－5x＋6 的系数，则 polyvalm(p,A)表示 A * A－5 * A＋6 * eye(size(A))，而 polyval(p,A)表示 A * A－5 * A＋6 * ones(size(A))。

【例 3-3】 已知多项式 $f(x)=2x^2+4x-5$，分别用 polyvalm 函数和 polyval 函数，求 $\boldsymbol{X}=\begin{bmatrix}2 & 4\\ 1 & 2\end{bmatrix}$ 的多项式的值。

程序代码：

```
X=[2 4;1 2];
p=[2 4 -5];
Y= polyvalm(p,X)
Y1 = polyval(p,X)
```

运行结果：

```
Y =
    19    48
    12    19
Y1 =
    11    43
     1    11
```

2. 多项式的根

一个 n 次多项式有 n 个根，这些根有的是实根，有的包含若干对共轭复根。MATLAB 提供了 roots 函数用于求多项式的全部根，其调用格式为

```
r=roots(p)
```

其中，p 为多项式的系数向量；r 为多项式的根向量，r(1)，r(2)，…，r(n)分别表示多项式的 n 个根。

MATLAB 提供了一个可由多项式的根求多项式的系数的函数 poly，其调用格式为

```
p= poly(r)
```

其中，r 为多项式的根向量；p 为由根 r 构造的多项式系数向量。

【例 3-4】 已知多项式 $f(x)=x^4+2x^3-4x+1$。

(1) 用 roots 函数求该多项式的根 r。

(2) 用 poly 函数求根为 r 的多项式系数。

程序代码：

```
p=[1 2 0 -4 1];
r=roots(p)
p1 = poly(r)
```

运行结果：

```
r =
  -1.6300 + 1.0911i
  -1.6300 - 1.0911i
   1.0000 + 0.0000i
   0.2599 + 0.0000i
p1 =
    1.0000    2.0000   -0.0000   -4.0000    1.0000
```

显然，roots 函数和 poly 函数的功能正好相反。

3.1.3 多项式部分分式展开

对分子多项式 $B(s)$和分母多项式 $A(s)$构成的分式表达式进行多项式的部分分式展开，表达式如下：

$$\frac{B(s)}{A(s)}=\frac{r_1}{s-p_1}+\frac{r_2}{s-p_2}+\cdots+\frac{r_n}{s-p_n}+k(s)$$

MATLAB 可以用 residue 函数实现多项式的部分分式展开，其调用格式如下：

```
[r,p,k]= residue(B, A)
```

其中，B 为分子多项式系数行向量；A 为分母多项式系数行向量；$[p_1; p_2; \cdots; p_n]$为极点列向量；$[r_1; r_2; \cdots; r_n]$为零点列向量；k 为余式多项式行向量。

residue 函数还可以将部分分式展开式转换为两个多项式的除的分式，其调用格式为

```
[B,A]= residue(r,p,k)
```

【例 3-5】 已知分式表达式 $f(s)=\dfrac{B(s)}{A(s)}=\dfrac{s+1}{s^2-4s+2}$

(1) 求 $f(s)$的部分分式展开式。

(2) 将部分分式展开式转换为分式表达式。

程序代码：

```
a=[1 -4 2];
b=[0 0 1 1]
[r,p,k]=residue(b,a)
[b1,al]=residue(r,p,k)
```

运行结果：

```
r =
     1.5607
    -0.5607
p =
    3.4142
    0.5858
k =
     []
b1 =
    1.0000    1.0000
al =
     1    -4     2
```

3.1.4 多项式的微分和积分

1. 多项式的微分

对于 n 阶多项式 $p(x)=a_n x^n+a_{n-1}x^{n-1}+\cdots+a_1 x^1+a_0$，其导数为 $n-1$ 阶多项式 $\mathrm{d}p(x)=na_n x^{n-1}+(n-1)a_{n-1}x^{n-2}+\cdots+a_1$。原多项式及其导数多项式的系数分别为 $\boldsymbol{p}=[a_1,a_{n-1},\cdots,a_1,a_0]$，$\boldsymbol{d}=[na_n,(n-1)a_{n-1},\cdots,a_1]$。

在 MATLAB 中，可以用 polyder 函数实现多项式的微分运算。polyder 函数可以对单个多项式求导，也可以对两个多项式的积和商求导，其调用格式如下：

```
p = polyder(p1)            % 求多项式 p1 的导数
p = polyder(p1,p2)         % 求多项式 p1 * p2 的导数
[p,q] = polyder(p1,p2)     % 求多项式 p1/p2 的导数，p 为导数的分子多项式系数，q 为导数的分子
                           % 多项式系数
```

【例 3-6】 已知两个多项式为 $f(x)=x^4+2x^3+2x^2+x+6$，$g(x)=2x^3-2x^2+1$。

(1) 求 $f(x)$的导数。

(2) 求 $f(x)*g(x)$的导数。

(3) 求 $g(x)/f(x)$的导数。

程序代码：

```
p1 = [1 2 2 1 6];
p2 = [2 -2 0 1];
p = polyder(p1)
poly2sym(p)
p = polyder(p1, p2)
poly2sym(p)
[p,q] = polyder(p2,p1)
```

运行结果：

```
p =
     4     6     4     1
ans =
4 * x^3 + 6 * x^2 + 4 * x + 1
p =
    14    12     0    - 4    36   - 20     1
ans =
14 * x^6 + 12 * x^5 - 4 * x^3 + 36 * x^2 - 20 * x + 1
p =
    - 2     4     8     0    28   - 28    - 1
q =
     1     4     8    10    20    28    25    12    36
```

2. 多项式的积分

对于 n 阶多项式 $p(x)=a_n x^n+a_{n-1}x^{n-1}+\cdots+a_1 x^1+a_0$，其不定积分为 $n+1$ 阶多项式 $i(x)=\frac{1}{n+1}a_n x^{n+1}+\frac{1}{n}a_{n-1}x^n+\cdots+\frac{1}{2}a_1 x^2+a_0 x+k$，其中 k 为常数项。原多项式和积分多项式分别可以表示为系数向量 $\boldsymbol{p}=[a_n,a_{n-1},\cdots,a_1,a_0]$，$\boldsymbol{I}=\left[\frac{1}{n+1}a_n,\frac{1}{n}a_{n-1},\cdots,\frac{1}{2}a_1,k\right]$。

在 MATLAB 中，提供了 polyint 函数用于多项式的积分。其调用格式为

```
I = polyint(p,k)   % 求以 p 为系数的多项式的积分,k 为积分常数项
I = polyint(p)     % 求以 p 为系数的多项式的积分,积分常数项为默认值 0
```

显然 polyint 是 polyder 的逆函数，即有 p=polyder(I)。

【例 3-7】 求多项式的积分 $I=\int(x^4+4x^3+2x^2-1x+6)\mathrm{d}x$。

程序代码：

```
p=[1 4 2 -1 6];
I= polyint(p)
poly2sym(I)
p= polyder(I)
syms k
I1 = polyint(p,k)
poly2sym(I1)
```

运行结果：

```
I =
    0.2000    1.0000    0.6667   - 0.5000    6.0000         0
ans =
x^5/5 + x^4 + (2 * x^3)/3 - x^2/2 + 6 * x
p =
     1     4     2    - 1     6
```

```
I1 =
[1/5, 1, 2/3, -1/2, 6, k]
ans =
x^5/5 + x^4 + (2*x^3)/3 - x^2/2 + 6*x + k
```

3.1.5 多项式曲线拟合

数据拟合的目的是用一个较为简单的函数 $g(x)$去逼近一个未知的函数 $f(x)$。利用已知测量的数据$(x_i,y_i)(i=1,2,\cdots,n)$,构造函数 $y=g(x)$,使得误差 $\delta=g(x_i)-f(x_i)$ $(i=1,2,\cdots,n)$在某种意义上达到最小。

一般用得比较多的是多项式拟合,即利用已知测量的数据$(x_i,y_i)(i=1,2,\cdots,n)$,构造一个 $m(m<n)$次多项式 $p(x)$:

$$p(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x^1+a_0$$

使得拟合多项式在各采样点处的偏差的平方和$\sum_{i=1}^{n}(p(x_i)-y_i)^2$ 最小。在 MATLAB 中,用 polyfit 函数可以实现最小二乘意义的多项式拟合。polyfit 函数求的是多项式的系数向量,该函数的调用格式为

```
p= polyfit(x,y,n)
[p,S]= polyfit(x,,n)
```

其中,p 为最小二乘意义上的 n 阶多项式系数向量,长度为 n+1; x 和 y 为数据点向量,要求是等长的向量; S 为采样点的误差结构体,包括 R、df 和 normr 分量,分别表示将 x 进行 QR 分解为三角元素、自由度和残差。

【例 3-8】 在 MATLAB 中,用 polyfit 函数实现一个 4 阶和 5 阶多项式,在区间[0,3]内逼近函数 $f(x)=2e^{-0.6x}\sin x$,利用绘图的方法,比较拟合的 4 阶多项式、5 阶多项式和 $f(x)$的区别。

程序代码:

```
clear
x = linspace(0,3 * pi,30);
y = 2 * exp( - 0.6 * x). * sin(x);
[pl,s1]= polyfit(x,y,4)
g1 = poly2str(pl,'x')    %将拟合后的多项式系数转换为字符形式
[p2,s2]= polyfit(x,y,5)
g2 = poly2str(p2,'x')
y1 = polyval(pl,x);
y2 = polyval(p2,x);
plot(x,y,'- * ',x,y1,':o',x,y2,': + ')
[p2,s2]= polyfit(x,y,5)
g2 = poly2str(p2,'x')
y1 = polyval(pl,x);
y2 = polyval(p2,x);
```

```
plot(x,y,'- * ',x,y1,':o',x,y2,': + ')
legend('f(x)','4 阶多项式','5 阶多项式')
```

运行结果如图 3-1 所示。

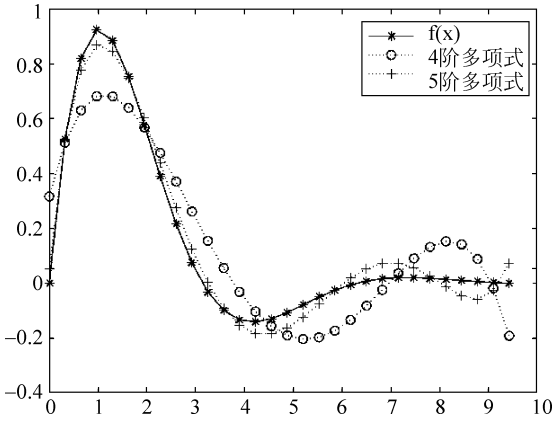

图 3-1　4 阶多项式和 5 阶多项式拟合 $f(x)$函数

由上述例题结果可知,用高阶多项式拟合 $f(x)$函数的效果更好,误差小,更加逼近实际函数 $f(x)$。

3.2　方程式

解方程在数学中是非常重要的,MATLAB 为方程的求解提供了强大的工具。符号方程分为代数方程和微分方程,下面介绍这两种方程的求解。

3.2.1　代数方程及代数方程组的求解

一般的代数方程包括线性方程、非线性方程和超越方程。当方程不存在解析解而又无其他自由参数时,MATLAB 提供了 solve 函数求取方程的数值解。命令格式如下:

```
solve ( 'eqn', 'v' )                    % 求方程关于指定变量 v 的解
solve( 'eqn1', 'eqn2',…,'v1','v2',…)  % 求方程组关于指定变量的解
```

说明:

(1) eqn 和 eqn1,eqn2,…是符号方程,可以是含等号的方程或不含等号的符号表达式。不含等号所指的仍是令 eqn＝0 的方程; v1,v2,…可省略,当省略时默认为方程中的符号变量。

(2) 其输出结果有 3 种情况:若单个方程有单个输出参数,则输出参数是由多个解构成的列向量;若输出参数和方程数目相同,则每个输出参数一个解,并按照字母表的顺序排列;若方程组只有一个输出参数,则输出参数为结构矩阵的形式。

【例 3-9】 使用 solve 求解下列两个方程组。

$$\begin{cases}\dfrac{1}{x}+\dfrac{1}{y}=a\\ \dfrac{1}{x}+\dfrac{1}{z}=b\\ \dfrac{1}{y}+\dfrac{1}{z}=c\end{cases} \quad 和 \quad \begin{cases}x^2+xy+y=0\\ x^2-4x+3=0\end{cases}$$

程序代码：

```
syms a b c x
eqn1 = 1/x + 1/y == a;
eqn2 = 1/x + 1/z == b;
eqn3 = 1/y + 1/z == c;
eqns = [eqn1 eqn2 eqn3]
S = solve(eqns)
```

运行结果：

```
S =
  包含以下字段的 struct:
    x: 2/(a + b - c)
    y: 2/(a - b + c)
    z: 2/(b - a + c)
```

```
syms a b c x
eqn1 = x^2 + x * y + y == 0;
eqn2 = x^2 - 4 * x + 3 == 0;
eqns = [eqn1 eqn2]
S = solve(eqns)
S.x
```

从工作空间可以看到，变量 S 是结构数组，因为方程的个数是两个而输出参数只有一个 S。若需求方程关于 x 的解，则可输出 S. x(S. x 是方程关于 x 的)解。S 与 S. x 的结果如下：

```
S =
  包含以下字段的 struct:
    x: [2×1 sym]
    y: [2×1 sym]
ans =
1
3
```

3.2.2 微分方程及微分方程组的求解

1. 使用 dsolve 函数求解

微分方程的求解比代数方程要稍微复杂一些，微分方程按照自变量的个数可以分为常

微分方程和偏微分方程，微分方程可能得不到简单的解析解或封闭式的解，往往不能找到通用的方法。MATLAB 提供 dsolve 函数来求解常微分方程的解析解（也称为符号解），命令格式如下：

```
dsolve('eqn','cond','v')                                    %求解微分方程
dsolve( 'eqn1 ,eqn2 ,…','cond1 ,cond2 ,…','vl , v2 ,…') %求解微分方程组
```

（1）求解微分方程组时，eqn 和 eqnl，eqn2，…是符号常微分方程，方程组最多可允许 12 个方程，在方程中，D 表示微分，D2、D3 分别表示二阶、三阶微分，Dy 表示 y 的一阶导数 dy/dx 或 dy/dt。

（2）cond 是初始条件，可省略，应写成“y(a)＝b，Dy(c)＝d”的格式，当初始条件少于微分方程数时，在所得解中将出现任意常数符 C1，C2，…，解中任意常数符的数目等于所缺少的初始条件数，是微分方程的通解；v1，v2，…是符号变量，表示微分自变量可省略，如果省略则默认为符号变量 t。

【例 3-10】 使用 dsolve 函数求解常微分方程，常微分方程为

$$(t^2-1)^2\frac{d^2}{dt^2}y(t)+(t+1)\frac{d}{dx}y(t)-y(t)=0$$

程序代码：

```
syms y(t)
eqn = (t^2 - 1)^2 * diff(y,2) + (t + 1) * diff(y) - y == 0;
S = dsolve(eqn)
S = dsolve(eqn,'ExpansionPoint', - 1)   %返回 t = - 1 周围的微分方程的级数解
```

运行结果：

```
S1 =
C2 * (t + 1) + C1 * (t + 1) * int((exp(1/(2 * (t - 1))) * (1 - t)^(1/4))/(t + 1)^(9/4), t,
'IgnoreSpecialCases', true, 'IgnoreAnalyticConstraints', true)
S2 =
t + 1
1/(t + 1)^(1/4) - (5 * (t + 1)^(3/4))/4 + (5 * (t + 1)^(7/4))/48 + (5 * (t + 1)^(11/4))/
336 + (115 * (t + 1)^(15/4))/33792 + (169 * (t + 1)^(19/4))/184320
```

【例 3-11】 使用 dsolve 函数求解微分方程组，微分方程组为

$$\begin{cases}\dfrac{dx}{dt}=2y\\[2ex]\dfrac{dy}{dt}=-3x\end{cases}$$

程序代码：

```
syms y(t) x(t)
[x,y] = dsolve('Dx = 2 * y,Dy = - 3 * x')
```

运行结果：

```
x =
(6^(1/2) * C2 * cos(6^(1/2) * t))/3 + (6^(1/2) * C1 * sin(6^(1/2) * t))/3
y =
C1 * cos(6^(1/2) * t) - C2 * sin(6^(1/2) * t)
```

2. 使用 ode 函数求解

前面介绍了如何利用 MATLAB 中的 dsolve 函数求解常微分方程，当求取精确解困难时，MATLAB 也为常微分方程提供了 7 种求解数值解的方法，包括 ode45、ode23、ode113、ode15s、ode23s、ode23t 和 ode23tb 函数，各函数的命令格式如下：

```
[t,y] = ode45(fun,ts,y0 ,options)    % 解常微分方程
```

说明：fun 是函数句柄或函数名；ts 是自变量范围，可以是区间[t0,tf]，也可以是向量[t0,…tf]；y0 是初始值，是和 y 具有同样长度的列向量；options 是设定微分方程解法器的参数，可省略，可以由 odeset 函数获得。

MATLAB 提供的 7 种数值求解函数的命令格式都与 ode45 函数的相似，功能有所不同，如表 3-1 所示。

表 3-1　常微分方程组 7 种数值求解的函数表

函数名	算　　法	适用系统	精度	特　　点
ode45	4/5 阶龙格-库塔法	非刚性方程	中	最常用的解法，单步算法，不需要附加初始值
ode23	2/3 阶龙格-库塔法	非刚性方程	低	单步算法，在误差允许范围较宽或存在轻微刚度时性能比 ode45 函数好
ode113	可变阶 AdamsPece 算法	非刚性方程	低-高	多步算法，误差允许范围较严时比 ode45 函数好
ode15s	可变阶的数值微分算法	刚性方程	低-中	多步算法，当系统是刚性时，可以尝试该算法
ode23s	基于改进的 Rosenbrock 公式	刚性方程	低	单步算法，可以解决用 odel5s 函数效果不好的刚性方程
odes23t	自由内插实现的梯形规则	轻微刚性方程	低	给出的解无数值衰减
ode23tb	TR-BDF2 算法，即龙格-库塔法，第一级采用梯形规则，第二级采用 Gear 法	刚性方程	低	对误差允许范围较宽时比 odel5s 函数好

刚性方程是指与常微分方程组的 Jocabian 矩阵的特征值相差悬殊的方程；单步算法是指根据前一步的解计算出当前解；多步算法是指需要前几步的解来计算出当前解。

【例 3-12】　使用 ode45 函数求解非刚性微分方程。

$$y_1''-1.2(1-y_1^2)y_1'+y_1=0$$

先将二阶微分方程式变换成一阶微分方程组：

$$y_1'=y_2$$

$$y_2' = -1.2(1-y_1^2)y_2 - y_1$$

程序代码：

```
function dydt = vdp1(t,y)
dydt = [y(2); 1.2 * (1 - y(1)^2) * y(2) - y(1)];
[t,y] = ode45(@vdp1,[0 20],[2; 0]);
plot(t,y(:,1),'-o',t,y(:,2),'-o')
% 绘制 y1、y2 图像曲线,y 的第一列与 y1 相对应,第二列与 y2 相对应
title('Solution of van der Pol Equation (\mu = 1) with ODE45');
xlabel('Time t');
ylabel('Solution y');
legend('y_1','y_2')
```

运行结果如图 3-2 所示。

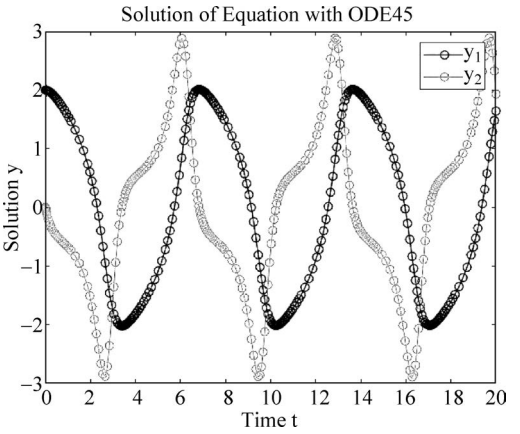

图 3-2　微分方程组解的曲线图

本章习题

1. 已知两个多项式：$f(x)=6x^4+4x^3+x+10$，$g(x)=7x^3+4x^2-3x+8$，求 $f(x)+g(x)$、$f(x)-g(x)$、$f(x)*g(z)$和 $f(x)/g(x)$的结果。

2. 已知多项式 $f(x)=6x^3+2x^2+7x+87$，分别求 $x=3.14$ 和 $x=[1,5,10,14,20,25]$的多项式的值。

3. 已知多项式 $f(x)=5x^2-8x+14$，分别用 polyvalm 和 polyval 函数求 $\boldsymbol{X}=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$的多项式的值。

4. 已知多项式 $f(x)=2x^4+x^3-3x+10$，用 roots 函数求该多项式的根 r，用 poly 函数求根为 r 的多项式系数。

5. 已知两个多项式：$a(x)=5x^4+4x^3+3x^2+2x+1$，$b(x)=3x+1$，计算 $c(x)=a(x)b(x)$，并计算 $c(x)$的根。当 $x=2$ 时，计算 $c(x)$的值，并将 $b(x)/a(x)$进行部分分式

展开。

6. 创建一个向量来表示多项式 $p(x)=3x^5-2x^3+x+5$，并对多项式求导。

7. 已知两个多项式：$f(x)=5x^4+3x^3+4x^2-3x-1$，$g(x)=6x^3-2x^2+x+7$，求多项式 $f(x)$的导数，求两个多项式乘积 $f(x)*g(x)$的导数，求两个多项式相除 $g(x)/f(x)$的导数。

8. 求多项式的积分，即求 $I=\int(5x^3+3x^2-8x+1)\mathrm{d}x$ 。

9. 已知 x 的取值范围为 0～20，计算多项式 $y=5x^4+4x^3+3x^2+2x+1$ 的值，并根据 x 和 y 进行二阶、三阶和四阶拟合。

10. 求解如下符号方程组的解。

$$\begin{cases}2a+3b+c=15\\a+2b+c=10\\7a+6b+4c=21\end{cases}$$

11. 求解如下符号微分方程的通解和当 $y(0)=2$ 时的特解。

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t)+y(t)\tan t=\cos t$$

12. 求解如下二阶常微分方程的通解，以及满足 $y(0)=0.4$，$y'(0)=0.7$ 的特解。

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}y(t)-y(t)=1-t^2$$